CyberGuarder: A Virtualization Security Assurance Architecture for Green Cloud Computing

Jianxin Li[1], Bo Li[1], Tianyu Wo[1], Chunming Hu, Jinpeng Huai[1], Lu Liu[2], KP Lam[3]

[1]School of computer Sci .& Eng.
Beihang University, Beijing, China
{lijx,libo,woty,,huaijp}@buaa.edu.cn

[2]School of Computing and Mathematics,
University of Derby, Derby, UK
l.liu@derby.ac.uk

[3]School of Computing and Mathematics
Keele University, Keele, Staffordshire,
k.p.lam@cs.keele.ac.uk

Abstract

With energy and power costs increasing as the size of IT infrastructures grows, virtualization technologies enable scalable management for large scale of virtual machines running on physical systems, and virtualization-based green cloud computing paradigm is springing up to provide a scalable and energy-efficient network software application (NetApp in short) supplement, consumption, delivery mode. However, the security problems will become more serious because data and infrastructures are fully shared among multi-tenant in a green cloud computing environment. Moreover security services generally affect the system's energy consumption and computing power. The success or failure of a practical application of a green cloud computing infrastructure strongly relies on its security solution. In this paper, we analyze the key security challenges faced by existing green cloud computing environments, and design a virtualization security assurance architecture named CyberGuarder to address the security problems with consideration of energy efficiency. In CyberGuarder, we provide three kinds of services from different security aspects. First, we propose a novel virtual machine security service incorporating a number of new techniques including 1) a VMM-based integrity measurement approach for a NetApp trusted loading, 2) a multi-granularity NetApps isolation mechanism for OS user isolation, 3) VM (Virtual Machine) isolation and virtual network isolation of multiple NetApps according to dynamic energy-efficiency and security needs. Second, we successfully developed a virtual network security service which provides an adaptive virtual security appliance deployment in the NetApp execution environment, and traditional security systems such as IDS, firewall etc. can be encapsulated into VM images and deployed into a virtual network in accordance with the utilization of virtualization infrastructure. Last, a security policy based trust management mechanism is proposed for access control to a resource pool and a trust federation mechanism across multiple resource pools to optimize the tradeoff between task privacy and computing cost requirements. We have studied these approaches in our iVIC platform, and some preliminary implementation experiments show that our approaches are effective and useful. Currently, we are building a virtual lab for our campus courses experiment based on our green cloud computing infrastructure iVIC, and CyberGuarder is an important virtualization security assurance system for the practical operation of iVIC platform.

Index Terms — Cloud Computing, Green Computing, Virtualization, Virtual Security Appliance, Security Isolation.

1 INTRODUCTION

Nowadays, Internet is evolving from the original communication tunnel (e.g., email) and content provider (e.g., Web) to an application center. More and more storage and computing capabilities are being delivered to end users through Internet. At the same time, numerous personal computers are used in the global world according to a recent Gartner report, and worldwide PC shipments have reached 82.9 million units in the second quarter of 2010, a 20.7 percent increase compared to the second quarter of 2009. In fact, enormous energy has been wasted due to idle resources. Our evaluating experiments on a Dell PC with Core2 CPU show that it consumes about 85W when sitting idle, almost half of the energy when sitting full-loaded, and a report [1] from NRDC showed that servers sitting idle still use 69-97% of total energy even if power management function is enabled, thereby computing has been a high-energy-consuming paradigm. With energy and power costs increasing as the size of IT infrastructures grow, holding expenses to a minimum is quickly becoming a top priority for many IT properties.

Recently, cloud computing paradigm [2][3] emerges to enable convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The virtualization-based cloud computing platforms also become popular to provide a new supplement, consumption, and delivery model for network software application (NetApp in short) based on the Internet. Computer virtualization refers to the abstraction of computer resources, such as the process of running two or more virtual computer systems on one set of physical hardware. The virtualization concept originated with the IBM mainframe operating systems of the 1960s. With virtualization, a system administrator could combine several physical systems into virtual machines on several computers as energy-efficient as possible, thereby unplugging the idle hardware to reduce power and cooling consumption. Moreover, virtualization can assist in distributing work so that servers are either busy, or put in a low power sleep state. The virtualization-based green cloud computing is leading to server consolidation and smaller electric bills, as well as heightened computer elasticity. Based on a cloud software operating environment, where virtualized, scalable and energy-efficient resource management approaches are provided to integrate loose-coupled resources, and improve their utility, users can be freed from the heavy work such as software deployment and maintenance. Many famous corporations such as Amazon, Google, Microsoft and Salesforce.com are becoming cloud service providers. iVIC[1] [4] is also a network software operating system to provide the elastic, scalable, and transparent resource management for NetApp, which can used to build a green computing infrastructure through its virtualization solution. It leverages virtual machine (VM) or virtual network to launch NetApp, and delivers desired software on-demand through presentation streaming mode (based on VNC) to PC or mobile phone (e.g., Android) with less energy consumption. Currently, iVIC has been used in the virtual course experiment system for our undergraduate and postgraduate students in Beihang University.

[1] http://portal.ivic.org.cn; http://ivic.org.cn/ivic/

However, the successful adoption of a virtualization-based green computing environment strongly depends on its security assurance mechanisms [5]. It should be noted that the computer security services generally impact the deployment and operation of the whole distributed system, and also usually be energy-consuming. Therefore, an integrated security solution not only can help the deployment of security services, but also can reduce its own energy consumption. To provide a secure NetApp operating environment, we identified three challenges that should be addressed, and proposed corresponding solutions which has been validated by using our iVIC platform.

First, a NetApp should be loaded without malicious tempering. Various malwares, such as virus, worms, Trojans and rootkits, continue to threaten the security of a VM. In particular, rootkit malware can hide its own process or disguise as a legal process, so as to escape the detection from virus scanner or intrusion detection system. The fundamental problem is that execution of malicious software or codes breaks the integrity of original computing system. Some typical integrity measurement approaches are Tripwire [5], IMA [7] PRIMA [8] and Google Chrome OS [11] etc. Their limitations are: ①. Most of them require to modify the OS kernel or applications (e.g., OS kernel in PRIMA need to be recompiled), and they cannot support legacy applications and close-box operating systems; ②. These systems are developed under the assumption that OS is secure. However, OS is susceptible to kernel attacks. For example, IMA is implemented through Linux kernel LSM, but LSM itself is easily suffered from bypass vulnerabilities; ③. Some approaches require the supporting of special hardware, e.g., Copilot [9] uses an add-in trusted PCI card to detect modifications to the OS kernel, and Google Chrome OS is based on a solid foundation of Ubuntu. When a Chrome OS is booted, it firstly checks the integrity of the OS through TPM to present the OS kernel from corruption or tampering of malware. Moreover, different NetApps should be isolated at different levels when some attacks occurred. Security isolation has been a key approach for the computing security, e.g., process isolation in time-sharing operating system is realized with virtual address space, and network isolation in the early network operating system is realized with firewall. It is critical to provide a multi-granularity isolation function because the NetApps can be downloaded from a third party (e.g, AppEngine, AppStore) and may have some malicious codes or faults. Amazon EC2 provides NetApp isolation with VM, Google AppEngine provides Java, Python binary application isolation, and VMware provides firewall-based network isolation. Some limitations are: ①. These approaches only were developed for special business purposes, which don't provide a generic approach for various NetApp isolation requirements; ②. The VM isolation cannot be adaptively adjusted according to the secure monitoring status of resource pool; ③. The firewall network isolation is only a packet filtering mechanism which relies on the physical network connections, and it cannot build any virtual network with various topologies, but also shared linking may suffer from tapping attacks.

Second, the network security systems (e.g., IDS, firewall) should be virtualized and easily deployed into a NetApp running environment. There are some related works such as *snort*, Cisco *Catalyst 6500 W/IDS* hardware etc. Some limitations are: ①. The deployment cost of a security system (e.g., IDS) is generally

high, and cannot be adaptively redeployed (e.g., Hardware IDS); ②. In place of the traditional security appliance, virtual security appliance has become a new way to be rapidly encapsulated and dynamically deployed in IT infrastructure. However, virtual security appliances are challenged with achieving optimal performance, as the physical resource is shared by several VMs. This issue is getting serious when a virtualized network intrusion detection system (NIDS) is deployed. ③. Virtual security appliances need to handle network traffic fluctuation and frequent network I/O which consumes lots of CPU cycles. They require an adaptive deployment mechanism to deal with dynamic computing capability requirements.

Last, a policy-based access control service should be used to protect the security of the virtual resources. Some NetApps often require scalable computing power, but a single resource pool (or private cloud) may not be able to provide enough resources for a large scale of users. Therefore, multiple resource pools sometimes need to be collaborated to achieve specific business goals. The oVirt[2] is built around *libvirt*, and provides a secure communication (GSSAPI/SASL2) and authentication mechanism (Kerberos/LDAP) for remote access to a resource pool. The OpenNebula[3] can build a hybrid cloud which extends a private cloud to combine local resources with resources from remote cloud providers such as Amazon EC2 or ElasticHosts. The limitations include: ①. Some approaches e.g., oVirt only provide a simple identity-based authentication mechanism without considering the real-time security policy updating and evaluation for the multi-tenant resource pool; ②. The existing approaches for hybrid cloud only provide an interface to invoke other public clouds, and they cannot support the federation of multiple pools. Therefore, they cannot solve the policy conflict problem for multiple pools federation; ③. The communication tunnel to the remote VM or virtual network also should be secured.

To address the above challenges, we propose a novel security assurance architecture named CyberGuarder, which enables the trusted loading of a NetApp, isolation of different NetApps, virtual security appliances for NetApp operating environment, and resource access control and remote accessing to NetApp. The major contributions are summarized as follows:

- We design a security assurance architecture named CyberGuarder for the NetApp operating systems, and the CyberGuader is a fully virtualization-based security solution for green cloud computing environments. At the same time, CyberGuarder is integrated into iVIC platform which is a virtual machine resource management system. Currently, we are developing a virtual lab for our students' practical course experiments in Beihang University, and CyberGuarder has played an important security assurance role for the operation of iVIC platform in the virtual lab.

- We design a virtual machine security service, which includes a software integrity measurement mechanism and a multi-level security isolation mechanism. The VMM-based integrity measurement approach named *VMInsight* can provide load-time and run-time monitoring for processes. VMInsight intercepts system calls and process behaviors by monitoring changes in the VM CPU register. It is im-

[2] http://ovirt.org/
[3] http://www.opennebula.org/

plemented in the hypervisor, which is completely transparent to the software and operating system running in the VM. The experimental results indicate that the performance overhead of VMInsight is less than 10% and energy consumption overhead is less than 5%. Anda multi-granularity NetApp sandbox mechanism is also proposed, and it can provide OS users isolation and VMs isolation based on available tools, and virtual network isolation solution ERVIN based on a layer-two tunnel VPN between distributed `vBridges`, and the meta-data such as virtual network topologies is maintained in a central node to optimize the traffics between VMMs.

- We design a virtual network security service, which provides an adaptive virtual security appliance deployment mechanism for a virtual network of NetApp running environment. To enable flexible network traffic detection, we design a dynamic *software mirror port* mechanism to control virtual network interface is under detected or not. The mirror port is implemented based on an Ethernet bridge configuration tool *brctl* to monitor the traffic. Moreover, we develop an online controller to adaptively control the distributed deployment of vIDS (a security appliance encapsulated the *snort*) according to various network topologies, traffic, energy and so on.

- We propose a virtual computing environment security service, which provides a policy-based trust management mechanism. The trust management mechanism not only provides a policy-based access control approach for a resource pool, but also provides a trust federation approach across multiple resources pools. To guarantee the real-time security policy updating and evaluation of the pool resource, we integrate the policy decision point and policy enforcement point with the virtual pool resource information service, and cache the status of access control list. The trust federation is implemented based on automated authentication procedure using the TrustVO federation policy, and users can directly access the VM or virtual network of another pool through secured VNC or VPN clients.

The rest of this paper is organized as follows. Section 2 elaborates the design of CyberGuarder, and the technology details and their performance evaluation results are presented in Section 2. We discuss related work in Section 3 beside Section 1. Finally, we conclude the whole paper in Section 4.

2 DESIGN OF CYBERGUARDER

According to the requirements analysis of network-based software operating system, we design the architecture of CyberGuarder in iVIC (shown in Figure 1). iVIC is a network computing platform based on distributed virtual resource container to encapsulate individual computing and storage devices so that they can provide virtualized entities, such as VMs or vDisks. Virtual machines are dynamically deployed and connected into virtual networks. Users may allocate their own virtual clusters or even complex virtual networks (vLabs) in iVIC to support hardware as a service (vHaaS) and software as a service (vSaaS) application scenario. In iVIC, software and hardware resources are organized in respective resource pools,

and software in software pool (SW Pool) can be downloaded and installed into VMs in hardware pool (HW pool) on-demand. There are four key security components in this architecture: *NetApp trusted loading*, *multi-level NetApp isolation*, *virtual security appliance* (e.g., vIDS), and *NetApp resource trust management*.

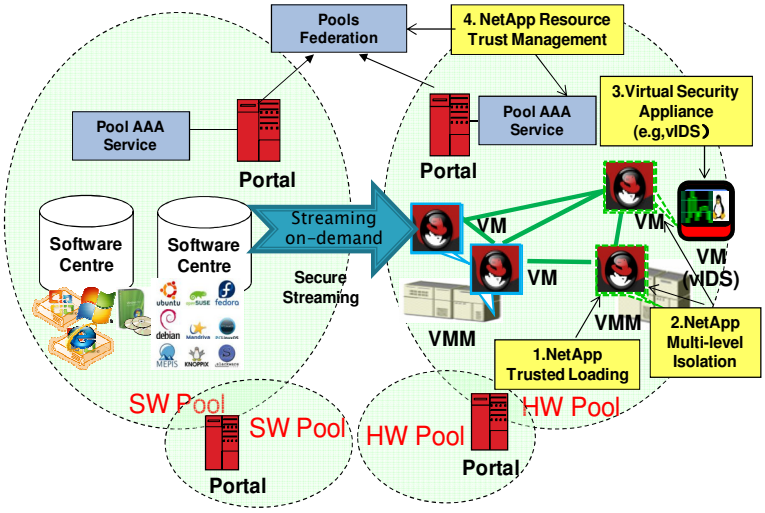

Fig. 1: The deployment architecture of CyberGuarder (AAA: Authentication, Authorization and Accounting. SW: Software, HW: Hardware, VM: Virtual Machine, VMM: Virtual Machine Monitor)

2.1 Virtual Machine Security Service

In this VM security service, we not only provide a VMM-based NetApp trusted loading approach-VMInsight, but also provide a multi-level security isolation approach based on the virtual machine technologies.

2.1.1 VMM-based Software Integrity Verification

In VMImsight, we leverage VMM-based system call interception approach to provide load-time and run-time integrity protection for a NetApp. Firstly, VMInsight intercepts and analyzes the system call sequence to identify and control the loading of software including *user applications*, *shared libraries* and *kernel modules*. Secondly, a system call correlation method is designed to establish the relationship of multiple system calls. Finally, VMInsight monitors the behavior of NetApp processes to recognize the malicious attacking patterns. For example, VMInsight can find hidden processes using the cross-view theory by collecting real VM process list and comparing them with that comes from the OS user's tool. The VMM-level protection mechanism ensures that the VM system can maintain its correctness and security even if guest OS kernel has been comprised. The VMInsight system also supports legacy or commodity guest operating systems, and it requires no modification to the guest OS.

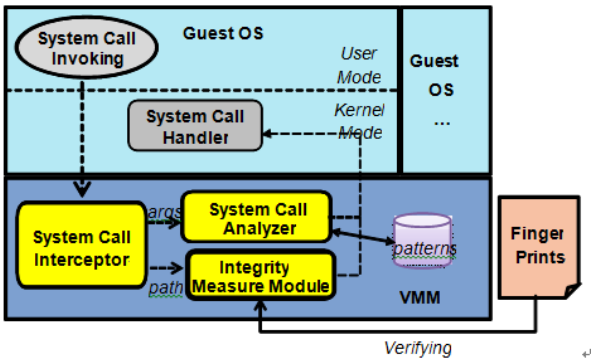

Fig. 2: the architecture of CyberGuarder VMInsight

The architecture of VMInsight is illustrated in Fig 2. VMInsight has three main components: *System Call Interpreter* (SCI), *System Call Analyzer* (SCA) and *Integrity Measure Module* (IMM). The VMInsight works as following two steps:

(1) SCI intercepts system call instruction (i.e. `INT 80h` or `sysenter`) invoked from the user mode in the guest OS, and identifies binary-executing related system call, resolves system call arguments to get executable path information.

(2) The arguments and path information are passed to SCA and IMM for further analysis. SCA analyzes the system call information based on configurable patterns to monitor the run-time behavior of processes. IMM receives executable paths from SCI, locates disk file using path information, and then measures file content. IMM takes measurements using the SHA-1 hash algorithm, and the fingerprint is then compared with known values stored in the fingerprint library.

Experiments: We have implemented VMInsight in two major VMMs (Qemu and KVM). We have conducted three experiment groups to evaluate its effectiveness to detect malicious processes, and the performance and energy-consumption overhead.

First, we use some malware samples to evaluate the effectiveness of VMInsight. We simulates malicious software's behavior of tampering with the already known software to test whether VMInsight can detect such integrity exception when `/usr/bin/ls` is loading. As shown in Fig. 3, VMInsight successfully found the integrity change of `/usr/bin/ls`. Next, we test VMInsight's capability of process monitoring using Apache Web server and some common-used applications. The results show that VMInsight can identify processes, detect network traffic, and monitor CPU usage and file operations. Such information will be exploited and integrated to identify malicious software behaviors. For example, the hidden processes which steal users' information can be located by analyzing the network packet receive/send status, thus resulting the report of malicious software.

PID	PPID	NAME	CPU%	SendBytes	RecvBytes	ReadBytes	WriteBytes	PATH	MD5	Trusted
1581	1573	rsyslogd	0%	12288	0	0	0	/usr/sbin/rsyslogd	6678abfd6d3d8a2a97bca3e0fcb4f984	YES
1593	1592	acpid	0%	16384	5	0	0	/usr/sbin/acpid	c25682102e4fe0d345923b55134939d5	YES
1613	1612	cron	0%	16384	0	76	0	/usr/sbin/cron	7149651c7dbc672e23f1ae0e655cf6f2	YES
1631	1626	apache2	0%	36864	104	0	0	/usr/sbin/apache2	066bc91b46a38490bf42b15c6c2ca454	YES
1632	1631	apache2	0%	0	0	0	0	/usr/sbin/apache2	066bc91b46a38490bf42b15c6c2ca454	YES
1752	1683	login	0%	18063	177	0	0	/bin/login	1e6966654edb9c8c106b989c55bc324a	YES
1763	1761	apt-get	89%	162977040	16863513	131	2048	/usr/bin/apt-get	141b1633992736135 59a799a20cfd3c5	YES
2839	2802	mysqld_safe	0%	95448	1008	0	0	/usr/bin/mysqld_safe	849762f87aa01e1af972b6a8205f8a17	YES
2845	1752	ls	0%	0	0	0	0	/usr/bin/ls	573ec4e8d3095cc33106c62cced6fc9e	NO

Fig. 3: System processes monitored by VMInsight

Second, we use some benchmark applications to evaluate the performance overhead of VMInsight for Qemu and KVM. As the results shown in Fig. 4, VMInsight incurs less than 10% performance overhead. According to the above analysis, we can conclude that the monitoring information provided by VMInsight can be used to develop a third-party security system.

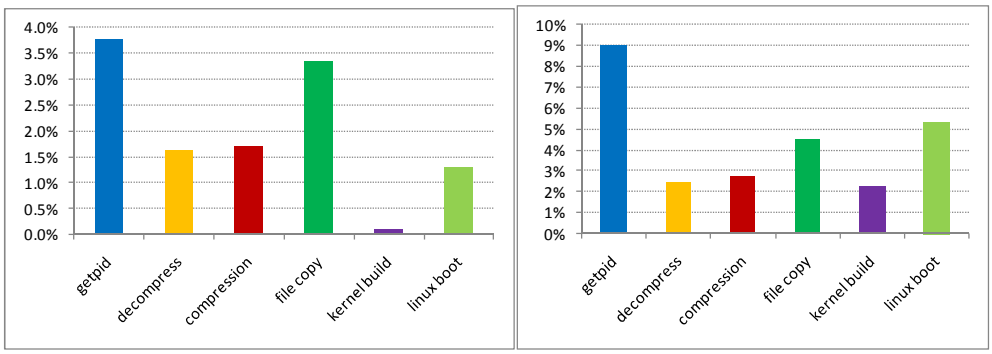

Fig. 4: Runtime overhead of VMInsight on Qemu (on the left) and KVM (on the right)

Finally, we use four benchmark applications to measure the energy-consumption overhead of VMInsight. We launch four different application tasks as shown in Table 1 on a `Dell OptiPlex 960` PC (with Inter(R) Core(TM)2 Quad 2.66GHz CPU, 4GB RAM and `Debian Linux` operating system) , and we use a `Voltech PM1000+` equipment to measure the power and energy consumption of this computer (exclude the monitor). The total energy consumption for each task under different execution environment configuration is listed in Table 2. Based on Table 2, we draw Fig. 5 to show the percentage of energy overhead for KVM and VMInsight compared with the physical machine. From this energy consumption experiment, we can reach two obvious results. First, if we just compare a KVM VM and a VMInsight service with a physical machine, it is obvious that the total consumption energy will increase because both the KVM and security processes will bring some extra overhead to the computer. The results show overhead incurred by VMInsight is less than 5% on KVM. Second, administrators need to pre-install a security monitor for each VM OS if there is no a virtualization layer security service. It will not only bring a heavy management burden, but also the utilization the security monitor software often be very low, so computer resources and energy are wasted. In particular, the energy consumption overhead incurred by the moni-

tor for every OS is generally 5%. If a security monitor is installed on each VM OS, the total energy wasted in a physical machine will significantly increase because a physical machine can generally run 20-30 Linux VMs. However, the VMInsight is only a module on the VMM layer, and it can serve for all VMs.

Table 1 The four experiment applications tasks

Application Task	Command String
Kernel Build	`make defconfig & make`
File Copy	`cp -r linux-source-dir otherwhere`
Compression (gz)	`tar zcf linux-source-dir`
Decompression (bz2)	`tar zxf linux-source.tar.bz2`

Table 2 The total energy wasted for each task (Watt*second)

Application Task	Physical Machine	KVM VM	KVM with CyberGuarder VMInsight
Kernel Build	197×10^3	295×10^3	304×10^3
File Copy	2.9×10^3	3.7×10^3	3.9×10^3
Compression(gz)	3.1×10^3	3.6×10^3	3.7×10^3
Decompression(bz2)	3.2×10^3	4.0×10^3	4.1×10^3

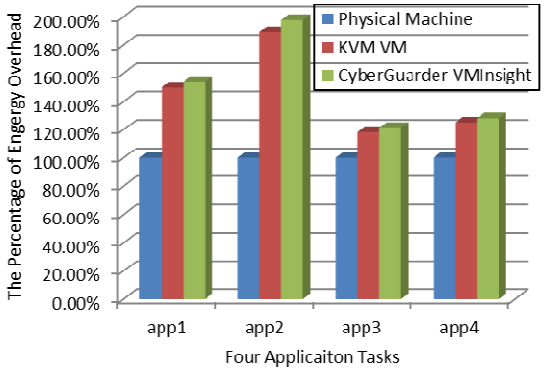

Fig. 5: The Energy consumption overhead for different tasks

2.1.2 Multi-granularity NetApp Sandbox Mechanism

Isolation is an important factor to improve the availability and security of applications running in a virtual environment is far superior to applications running in a traditional, non-virtualized system. While virtual machines can share the physical resources of a single computer, they remain completely isolated from each other as if they were in separated physical machines. If, for example, there are four virtual machines on a single physical server and one of the virtual machines crashes, the other three virtual machines remain available.

As shown in Fig. 5, we design a multi-granularity NetApp sandbox mechanism in CyberGuarder, which can provide security isolation at different levels. The isolation at the user and application levels is

achieved with existing tools, and the virtual network level isolation is achieved with CyberGuarder ERVIN. In response to the user isolation requirement in an OS, we use *chroot* to create and host a separate virtualized copy of the software system. Now, we are also adopting Linux kernel *seccomp* to allow processes to call a very small subset of system calls, e.g., `read`, `write`, `sigreturn`, and `exit`. For the NetApps isolation requirement among VMs, we assign security policies for a resource pool and a scheduler (deployed with the Web Portal) can automatically deploy VMs according to the NetApps isolation policies. For the virtual network isolation requirements, we design the ERVIN which uses a layer-two tunnel VPN between distributed `vBridges`, and the meta-data such as virtual network topologies is maintained in a central node to optimize the traffics between VMMs. CyberGuarder ERVIN provides a data transmission mechanism in a P2P manner for virtual network, the network packets between different virtual machines do not transit through a central server, so make full use of the network bandwidth between the hosts to improve the efficiency of virtual machines network packets transmission.

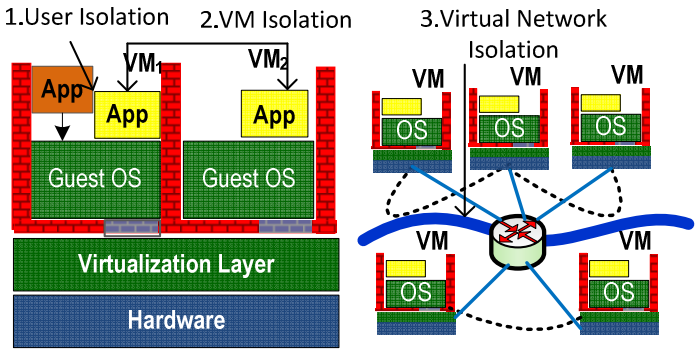

Fig. 6: Three-level NetApp isolation in CyberGuarder

Experiment: In order to evaluate the performance of CyberGuarder ERVIN can take advantage the network bandwidth between hosts compared with OpenVPN-based virtual network approach, we design an experiment to measure their performance. We test the performance of virtual networks connected with 2, 6 and 10 virtual machines respectively, and each virtual machine is deployed on different hosts, and we use NetPIPE to measure the network throughput. In Fig. 7, we plot the network throughput against the size of packet sent between VMs by NetPIPE. As shown in Fig.7, CyberGuarder ERVIN has a better performance compared with OpenVPN on both communication throughput and scalability. There is no obvious performance degradation occurs when increasing the number of virtual network peers (because the experimental virtual machines lie on different hosts), while the OpenVPN performance is much affected by the scale of virtual network and communication overhead. It drops to 48% of the maximum speed when the number goes up to 5.

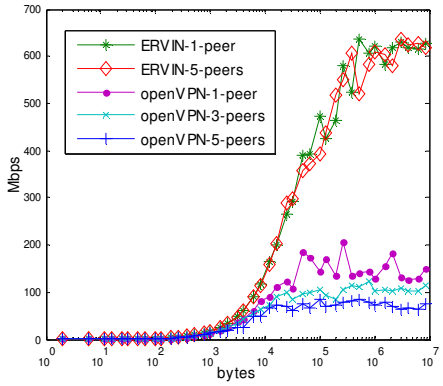

Fig. 7: The performance of CyberGuarder ERVIN vs. openVPN

2.2 Virtual Network Security Service

In CyberGuarder, we also design an adaptive security system deployment mechanism for virtual network environments. We encapsulate traditional network security systems into virtual security appliances and adaptively deploy them into virtual networks to safeguard the applications running in the virtual networks. We also design a dynamic provision approach based on fuzzy control theory, which can continuously control resource allocation for virtual security appliance to deal with varying network traffic while still satisfying the performance or energy consumption requirements.

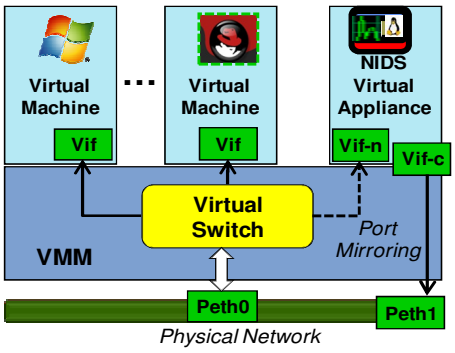

Fig. 8: The architecture of CyberGuarder vIDS

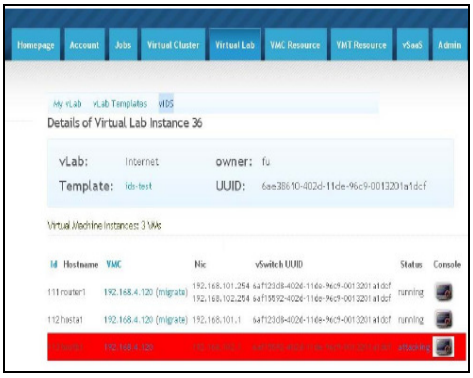

Fig. 9: A snapshot of vIDS demo in iVIC

As shown in Fig. 8, VMs `Vifs`, `Peth0` and the `Vif-n` of vIDS are connected with a virtual switch. `Peth0` is a physical network interface, and all packets of a VM will go through `Peth0`. Virtual NIDS has two virtual network interfaces: `Vif-n` and `Vif-c`. The `Vif-n` is connected with the mirror port of the virtual switch, and duplicates and forwards monitored packets to the vIDS. To guarantee no disturbance to the whole system, a physical network interface `Peth1` is dedicated for `Vif-c` to connect with physical network. Linux bridge works in layer 2 protocol, and acts in a similar manner with physical switch, so we choose Linux bridge as a virtual switch in our implementation. We have slightly modified Linux bridge to support port mirroring, where a flag is added to `net_bridge_port struct` to indicate whether the network traffic traversing this port will be duplicated and forwarded or not, and a pointer is added to

`net_bridge struct`, it points to a bridge port to indicate that this port is the mirror port of the bridge. This is a very flexible approach to integrate with available network intrusion detection systems. Any port can become a mirror port. We can dynamically configure which virtual network interface is under detection and which is not. We have added four commands to brctl[4], `add_mirror_port`, `del_mirror_port`, `add_src_if`, and `del_src_if`. To enable the port mirroring function, we first need to execute "`brctl add_mirror_port <ifname>`" to assign a bridge port to be the mirror port, and any packet forwarded to the mirror port will be sent to the virtual network interface connected with this port. Next, we call "`brctl add_src_if <src_if_name>`" to specify that the packets flow through "`src_if_name`" interface will be duplicated and forwarded to the mirror port. If we want to cancel the monitoring of one interface, we can run the command "`brctl del_src_if <ifname>`". Finally, we use "`brctl del_mirror_port <ifname>`" to turn off the port mirroring function.

Fig. 9 also shows a snapshot of our vIDS demo in iVIC portal. The iVIC Portal is deployed as the user interface and provide management console for a virtual machine pool. In iVIC Portal, users can create a virtual cluster or virtual lab with some complex virtual networks. In this figure, a virtual lab instance is created with vIDS service, and if a virtual machine is detected be attacked, the attacked virtual machine will be highlighted in red.

Experiment: We have implemented CyberGuarder vIDS which is based on snort. To evaluate its packet analysis performance and the power changing with the time-varying workloads, we have conducted two experiments. Fig.10 shows the simulated workload of 160 seconds, we change the packets sending speeds every 10 seconds.

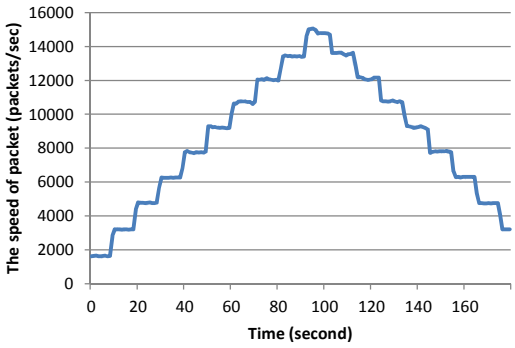

Fig. 10: The simulated workload for vIDS

We launch vIDS to evaluate the effectiveness of our dynamic provision approach. Fig.11 shows the transient and accumulated drop rate for 2% MPDR (Maximum Packet Drop Rate). We can see that the transient drop rate fluctuates up or down at the MPDR, while the accumulated drop rate tends to gradually converge at the MPDR. The results show that our system can precisely allocate resource for NIDS ac-

[4] A user-mode tool for controlling Linux Bridge.

cording to its resource demands, while still satisfying the performance requirements of NIDS.

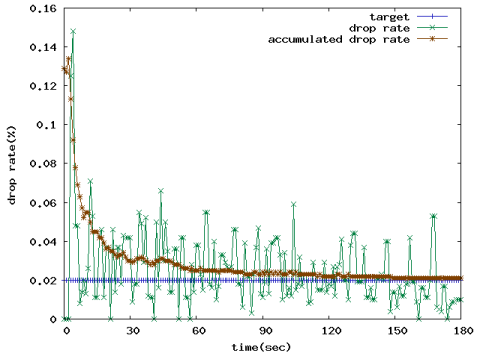

Fig. 11: Transient and accumulated packet drop rate for 2% maximum packet drop rate

At the same time, we also use Voltech PM1000+ equipment to measure the power and energy consumption of this computer (exclude the monitor). From the Fig.12, we can see the power of CyberGuarder vIDS (encapsulating a snort IDS) quickly reduces with the decreasing of network workload. But the power of common vIDS reduces very slowly. This is because that CyberGuarder vIDS has the capability to dynamic control the CPU usage based on the workload of the packet receiving, but the common vIDS has not.

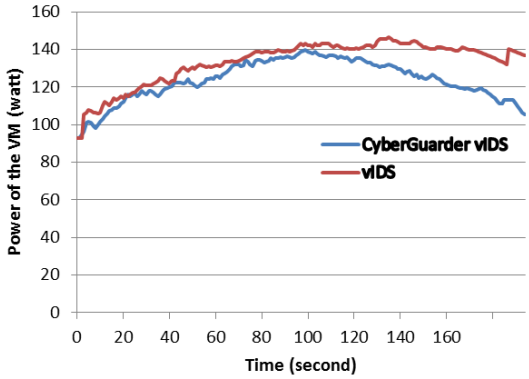

Fig. 12: The Power of CyberGuarder vIDS and common vIDS

2.3 Virtual Environment Security Service

- **Policy-based security service for the local VM pool**

In a VM pool, many physical machines are connected in a high-speed network, and every physical machine can run several virtual machines simultaneously. In this pool, users can create their own virtual clusters or virtual labs by connecting assigned virtual machines. The security policy for this virtual machine pool is configured in a centralized portal, and it authenticates the user's identity, and manages the access control policies of virtual machines.

As shown in Fig.10, the steps of security policy enforcement in a VM pool is as follows: (1). The user

firstly login the virtual machine pool with its password or certificate; (2). The authentication server in the VM pool verifies the identity of the login user; (3).User creates a virtual cluster or a virtual lab on the portal workspace; (4) When the user performs a task involving some operations on the virtual machine pool, these actions need to be authorized firstly by the user policy server. If all the actions involved in this task are permitted, then this task is submitted to the scheduler; (5) The portal submits a description file of user's task to the scheduler; (6) The scheduler deploys the virtual machines according to the description of task file and pool information service. After a virtual cluster or a virtual lab is deployed in the VM pool, then the user can access directly related virtual machines in this pool via remote client tools. The procedures are: (a1). The user firstly requests a proxy credential or certificate from the portal after an automated authentication procedure; (a2). The user can access the virtual machine through SSH or VNC clients.

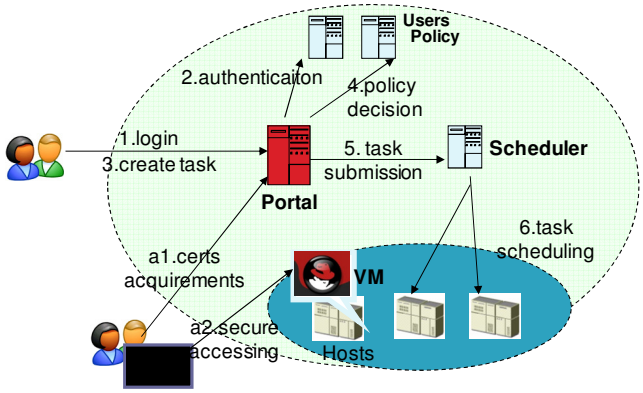

Fig. 13: The architecture of security policy enforcement in a local VM pool

On the side of policy server, the policy is stored with format of *policy*=`(subject, object, constrains set)`, where the `subject` is a user, the `object` can be a VM pool or a physical machine, and the `constraints set` includes operation constraints on the VM pool and physical machines. The possible operations for a VM pool or physical machines are listed in Table 1, and the constraint variables with the operations are listed in Table 2.

Table 3 The operations for VM pool and physical machine

Object	Operation	Statement
VM pool or physical machine	Any	A wildcard that represents any operation on the VM pool
VM pool	ivic#createVCluster (VMCount)	Create a virtual cluster with a variable that specifies the number of virtual machines in this cluster
VM pool	ivic #createVLab VMCount, VSwitchCount)	Create a virtual lab with two variable that specify the number of virtual machines and the number of virtual switches in this cluster
physical machine	ivic #deployVM	The operation to deploy a virtual machine into the VM pool
physical machine	ivic #deploySwitch	The operation to deploy a virtual switch into the VM pool
physical machine	ivic #startVM	The operation to start a virtual machine deployed in the physical machine
physical machine	ivic #startSwitch	The operation to start a virtual switch deployed in the physical machine

Table 4 The constrain variables with the operations for VM pool and physical machine

Constraints Variable	Statement
vLabCount	The maximum number of virtual labs that user can create
vClusterCount	The maximum number of virtual clusters that user can create
vSwitchCount	The maximum number of virtual switches that user can create
liveSwitchCount	The maximum number of virtual switches that a physical machine can run
liveVMCount	The maximum number of virtual machines that a physical machine can run

An example of the security policy is as follows:

```
(Alice@ivic.org.cn, pool-1, [vClusterCount ≤2]);
(Alice@ivic.org.cn, Host: 192.168.0.119, [liveVMCount≤2]).
```

This policy means that Alice mostly can create two virtual clusters on VM pool-1, and specially can mostly start two virtual machines on the physical machine 192.168.0.119 simultaneously.

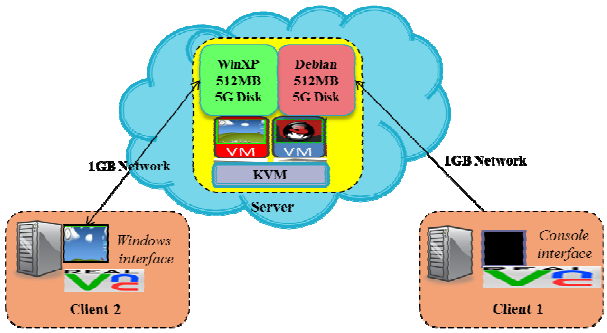

Fig. 14: Remote VM access through VNC with TLS and without TLS

Experiment: Because the cloud is a centralized infrastructure, all VMs are executed on the server side. If a client wants to interact with its VM, it must connect this VM based on remote display tool (e.g., VNC used in CyberGuarder). However, if too many clients connect their VMs located in a resource pool, the bandwidth will become one issue. Therefore, our security mechanism should guarantee not to bring too much traffic overhead. In this experiment, we measure the network traffic. As shown in Fig. 14, we configured a test environment on QEMU VNC Server for two virtual machines and a RealVNC viewer, and with 1 GB network connection. Then, we measure the communication throughput when the channel from VNC desktop to the VNC server is encrypted or not. On the test server physical machine, the network interface cards of virtual machines is bridged to the physical network of a computer. The experiments are divided into two groups, one runs Debian Linux on a virtual machine with a console, and a simple '`ls`' shell command is executed to continually refresh the screen, and another runs Windows XP OS on a virtual machine with a Windows desktop, and the Media Player is launched to play video with a full-screen mode on a desktop of 1024 x 78 resolution. On the client side, we access these two virtual machines through RealVNC client with TLS encryption and without encryption respectively.

The results are shown in Fig.15 and Fig 16. Fig. 15 indicates that average network traffic is about 500 Kbps for a console client, and the security mechanism only add little network traffic. The Fig. 16 indicates

that the two modes almost have the same network traffic (about 2000 Kbps) because the encryption brings lower percentage to the larger total network traffic when a Windows desktop is transferred.

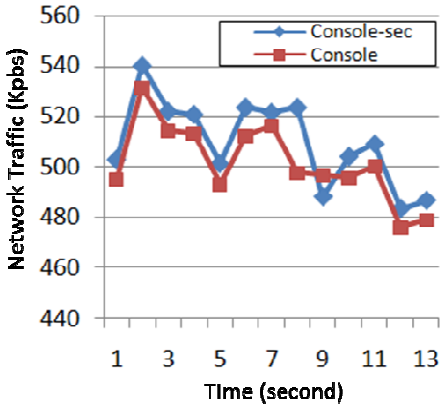

Fig. 15: Network traffic of communication using console client with TLS and without TLS

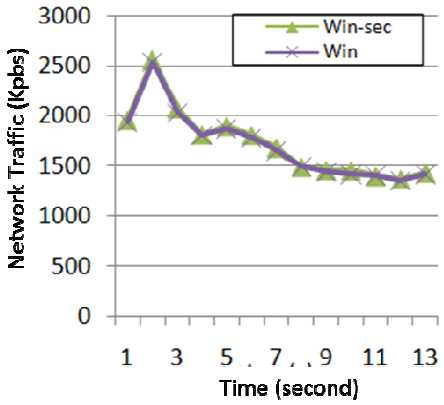

Fig. 16: Network traffic of communication using Window client with TLS and without TLS

- **Policy-based trust federation for multiple VM pools**

In general, a company or organization can build its own private resource pool (private cloud), while the resource capacity of a private pool cannot fulfill all business requirements. For example, the required computing resources of an application in Facebook ever increased to 3000 hosts from 50 hosts in only three days. In CyberGuarder, we design a policy-based trust management service named TrustVO (shown in Fig. 17) to improve its scalability by federating multiple resource pools (clouds). The security policy is specified by role mapping, and possible conflicts are resolved by our existing work PEACE-VO [10].

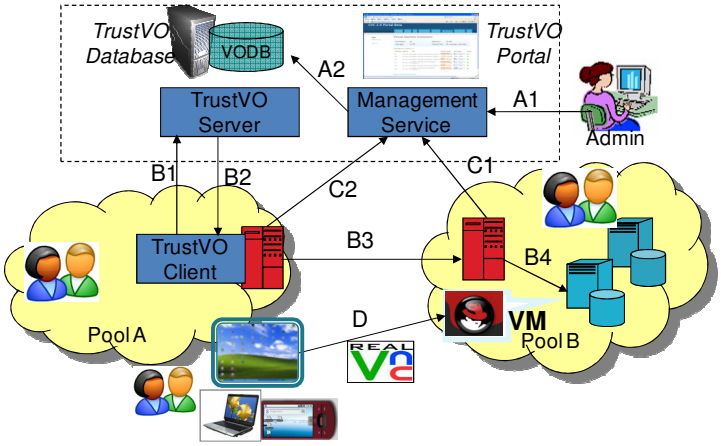

Fig. 17: The management and authentication workflow of CyberGuarder TrustVO.

If some pools need to be federated, the management steps of TrustVO administrator are as follows:

- A1: The *Admin* logins into the *TrustVO Portal*;
- A2: The *TrustVO Server* authenticates the identity of *Admin* and executes the management for *VO*

Database via *Management Service*.

When a client from *Pool A* wants to use resources provided in *Pool B*, the authentication steps of TrustVO are as follows:

- B1: The *Pool A* sends VO membership credential requests to the *TrustVO Server*;
- B2: The *TrustVO Server* generates a credential according to the requests;
- B3: The *Pool A Portal* sends a job scheduling request to another *Pool B Portal*;
- B4: The *Pool B* makes authorization decision for the job scheduling across pools (clouds);
- C1, C2: The information of federated virtual resource is reported to the *Management Server*, and clients can query the resources status through accessing the *Management Server*;
- D: The user can access any authorized VM Desktop via RealVNC.

Experiment: To provide a general trust federation approach to interoperate with the existing cloud security infrastructures. In CyberGuarder TrustVO, an administrator can configure different security communication mechanisms. Currently, CyberGuarder can support two major security communication toolkits, GSI SOAP and OpenSSH. The SOAP security mechanism has been extensively used in Grid security infrastructure and some SOA security services. The OpenSSH is also a traditional remote secure access tool and can be easily used. We use these two security communication mechanisms for TrustVO respectively. As shown in Table 5, the average authentication time is about 350 ms for SOAP security and 380 ms for the OpenSSH RSA mechanisms, and there are no obvious differences between these two mechanisms.

Table 5 Authentication time with different security communication mechanisms

	SOAP Security (WS-Security)	OpenSSH RSA
Authentication Time (ms)	350	380

3 RELATED WORK

In 1970s, S.E. Madnick and J. J. Donovan [12] from MIT, who had engaged in research work relevant to IBM VM/370, firstly put forward the idea to improve the system security based on virtual machine isolation mechanism. Many years later, the virtualization technology began to receive attentions again with the prevalence of new Internet-based computing paradigms e.g., Cloud computing.

3.1 Virtualization Security

In 2008, Kevin Borders et al. from Michigan University summarized some security technologies related to virtualization. This is an earlier overview and analysis to the security mechanisms of the virtual machine [13], and it introduces related work including intrusion detection, intrusion defense and the honeypot system based on virtualization technology and so on. Nowadays, the virtualization technology has been brought to the forefront in the area of industry and manufactory, and it also becomes an important

technology to build a green IT infrastructure. However, how to ensure its security becomes a bottleneck of its real adoption. In 2009, 11 scholars from UC Berkeley Reliable Adaptive Distributed Systems Laboratory published a report on cloud computing [14]. In this report, they give a concept model and some research trends of cloud computing. Especially, the top 10 problems to cloud computing are discussed, and three of them are related to the security issues. To build a secure execution environment for the network software applications, the related work on virtualization can be classified into three types: security isolation, trusted loading, and monitoring & detection.

Security Isolation:

In the nearly 40 years of virtualization technologies, virtual machine is used from the previous physical environment isolation to dynamic business logic isolation, and virtualization computing system realizes the balance and synthesis of multiple functions, such as computing performance, application efficiency, security isolation and so on. At the same time, we are suffering from more system vulnerabilities and network attacks occurred frequently. An important motivation of early IBM VM/370's appearance is to realize partition isolation, and the VMM (Virtual Machine Monitor) technology makes it possible to create a lot of virtual machines to run independent operating systems in the same physical hardware. The virtualization system can avoid the system information leakage, which may be caused by users' improper or malicious operations. Yan Wen [15] has presented a virtualization isolation model, and he put forward a new kind of virtual machine model based on the hardware abstraction layer virtualization (Safe Virtual Execution Environment, SVEE), which implements the Bell-LaPadula confidentiality model and Biba integrity model. Researchers from MIT provides Flume system [15] which is used for distributed information flow control, managing the data flow of application segment, realizing the integrity and privacy of data. However, the cloud is general a multi-tenant computing environment, and an isolation solution at different levels is required. In CyberGuarder, we achieve this goal through lightweight application isolation, virtual machine isolation and virtual network isolation.

Trusted Loading:

A fundamental reason of unreliable system is because that the integrity of systems is break by malicious software or codes. Therefore, how to ensure the software origins from a trusted party is an effective way to guarantee the intrinsically security of a system. The integrity measurement is a way to prove that the providers and sources are reliable and accountability, that means the software files have not been damaged or tampered. The researchers from IBM propose the HIMA [17] which also employs a VMM-based approach to take integrity measurements on user programs and kernel codes. However, HIMA needs modification to the guest OS. Arvind Seshadri et al. from CyLab of CMU design a light-weighted hypervisor named SecVisor which can ensure the integrity of kernel code of Linux and prevent malicious code injection and so on. In a word, the major way of SecVisor is to virtualize the MMU and IOMMU. However, SecVisor is a light-weighted hypervisor, and it can only operate one guest OS and cannot be used when there are hybrid memory pages.

Security Monitoring and Detection:

Monitoring technology is another key way to keep the system running healthily. The VMM has a good introspection capability, thereby it's intensively used to monitor the device status and attack behaviors. The research work on this direction can be classified into two types: one is a pure monitor function for the memory, disk and I/O of virtual machines, and another is the system security detection including the malicious attacks and some intrusion behaviors. B.D.Payne and Wenke Lee propose XenAccess library [18] which was based on the Xen 3.0's existing XenControl library and Blktap arch. XenAccess mainly aims to monitor the virtual machine memory and disk I/O, and this approach could be easily extended to monitor network flow and CPU. But XenAccess needed to be deployed to Xen Domain0. B.D.Payne et al. further propose Lares [19] which realized an active monitoring function based on Xen and Window XP. Tal Garfinkel [20] from Stanford University studied virtualization's characteristics such as mobility, security monitoring etc., and he propose some approaches for intrusion detection, integrity checking and honeypot system and forensics based on the monitoring capability of VMM layer. For example, a VMM introspection based architecture for intrusion detection is used to analyze the attackers' behavior. KvmSec [21] is an extension of Linux Kernel Virtual Machine (KVM), which can prevent KVM from being attacked by virus and kernel rootkit. KvmSec provides a transparent way to data collection and analysis to the guest OS. Peter M. Chen et al. [22] from Michigan University propose a detecting past and present intrusions method through vulnerability specific predicates. This method can monitor the internal running state of virtual machine based on the introspection capability.

Besides, many IT companies such as Amazon, Google, and Microsoft have launched their cloud computing and green computing projects, and the virtualization security is also a major product. In the Amazon S3 storage service, the owner can assign access control policy to specify who can read or write or have other privileges. In the Amazon EC2 computing services, unauthorized access to the virtual machine or virtual network can be prevented through a firewall policy configuration on IP and routing. VMWare VirtualCenter is a kind of task-based privilege management system, which is used to control the permissions of administrator and users on the platform. The system administrators can assign the user permissions through configuration of user/group, roles to tasks. VMWare vShpere is a cloud operating system including VMsafe and VMWare vShield Zones, and they provide firewall, anti-virus, intrusion detection and intrusion prevention capabilities to the virtual environment. VMware vShield Zones can configure VLAN to separate the network and create a security boundary. Microsoft Hyper-V provides some security functions based on virtual machine, such as malicious code execution prevention, role-based access control, and streamlined system architecture. RedHat oVirt integrates the user and data management based on LDAP, and distributes the user ticket based on Kerberos infrastructure, and uses freeIPA project to implement virtual resource authentication, authorization and accounting. CyberGuarder is a much different solution compared with these products, and three kinds of security assurance services on the granularity of virtual machine, virtual network and virtual pool, and the security mechanisms can be smoothly

integrated the virtualization infrastructure and interoperated with the existing local security infrastructures.

3.2 Green Cloud Computing

The power consumption of computers and data centers is growing with unprecedented levels: the EPA estimates U.S. data centers will consume 100 billion kilowatt hours in 2011. Much of this energy is wasted in idle systems: in typical deployments, server utilization is below 30%, but idle servers still consume 60% of their peak power draw. In recent years, there are many research efforts focused on how to achieve the energy-saving for computers. The traditional energy saving approaches for a computer includes CPU and storage equipment improvements, the power management and dynamic voltage scaling (DVS) technologies based on operating systems. For instance, David Meisner et., al propose the PowerNap [24], which is an energy-conservation approach where the entire system transitions rapidly between a high-performance active state and a near-zero-power idle state in response to instantaneous load, and a power provisioning approach provides high conversion efficiency across the entire range of PowerNap's power demands. Kephart et al, [25] build a framework with consideration to both the power management and performance management, and use two existing IBM products, one that manages performance and one that manages power through dynamic voltage scaling (DVS) approach, and resulting in power savings of approximately 10%.

The traditional energy-saving approaches are mainly based on operating systems with full knowledge of and full control over the underlying hardware, but the distributed nature of multi-layered virtual machine environments makes such approaches insufficient. Cloud virtualization can significantly improve efficiency by leveraging the utilization and consolidation of virtual machines with minimum number of powered physical machines. Obtaining energy efficiencies in data centers is highly specialized and capital intensive. In 2009, Francis and Richardson [26] present a green maturity model for virtualization, and focus on reduction in energy consumption over the full equipment life cycle as the prime motivator for "green" application design. Some researchers [27] from IBM have presented a server workload analysis for power Minimization user consolidation to reduce the datacenter energy. The basic principal is turn on/off the server according to specific policies. Jan Stoess et al. [28] present a novel framework for energy management in modular, multi-layered operating system structures. This framework targets hypervisor-based virtual machine systems, and the guest level energy management relies on effective virtualization of physical energy effects provided by the VMM.

With the virtualization technologies are extensively studied, and the security services should be provided. However, the security services generally will add some extra energy consumption, and there are few related work consider much this issue [29].CyberGuarder designed in this paper is an important enhancement to the security of a green cloud computing environment. First, security is an important foundation to enable the green cloud computing infrastructure can be actually deployed and applied. Second, the CyberGuarder itself provides security service based on virtualization technology in different grained

level with many energy-saving benefits of virtualization. For example, some security service is deployed in virtual machines, and it can also be dynamically deployed or consolidation and can sleep or shut down when the environment risk is low. Finally, we also provide some energy-aware approaches into the security services policy, thereby administrators can dynamically deploy or control the security services according to the energy or performance requirements.

4 CONCLUSION

Computing is not only a high-tech one, but also a high-energy-consuming one. Because there are too many idle personal computers which waste a lot of energy, many researchers are seeking a new computing paradigm to realize green computing. Some centralized-based computing approaches based on virtualization technology, e.g., cloud computing, emerged to improve the efficiency and availability of IT resources and applications through virtualization. These approaches are eliminating the old "one server, one application" model and it becomes a trend that multiple virtual machines run on a physical machine. However, the security problem is becoming a barrier of virtualization technology in an open Internet environment. Therefore, the security issues will be more serious in an open NetApp operating system. At present, we are working on our ongoing virtualization project iVIC which aims to build a reliable and scalable NetApp operating system. We proposed an security assurance architecture named CyberGuarder, which provides NetApp virtual machine security, virtual network security and virtual environment security services. Some detailed approaches such as integrity verification, multi-level NetApp isolation, virtual security appliance (e.g., vIDS), and NetApp resource trust management services have been discussed in detail in this paper. Currently, the CyberGuarder has been used to build a secure virtual lab environment in our campus of Beihang University. Our future work is to further improve the reliability of a virtual environment, and the availability of data storage, virtual machine and virtual network. For example, we are implementing a live distributed snapshot technology for virtual networks, which can dynamically create a snapshot of the virtual network, and migrate the whole virtual network into another virtual machine pool and recover it quickly.

REFERENCES

[1] Natural Resources Defense Council "Recommendations for Tier I ENERGY STAR Computer Specification",.http://www.energystar.gov/ia/partners/prod_development/revisions/downloads/computer/RecommendationsTierICompSpecs.pdf

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," 2009.

[3] Gabriel Mateescu, Wolfgang Gentzsch and Calvin J. Ribbens.Hybrid Computing—Where HPC meets grid and Cloud Computing, 2010.(in press) doi:10.1016/j.future.2010.11.003

[4] Yang Chen, Tianyu Wo, Jianxin Li, "An Efficient Resource Management System for On-Line Virtual Cluster Provision," cloud, pp.72-79, 2009 IEEE International Conference on Cloud Computing, 2009

[5] Dimitrios Zissis, Dimitrios Lekkas.Addressing cloud computing security issues. Future Generation Computer Systems,2010.(in press) doi:10.1016/j.future.2010.12.006

[6] G. Kim and E. Spaord. The Design and Implementation of Tripwire: A File System Integrity Checker. Purdue University, November 1993.

[7] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, “Design and implementation of a tcg-based integrity measurement architecture,” in Proceedings of the 13th USENIX Security Symposium, August 2004.

[8] T. Jaeger, R. Sailer, and U. Shankar, “Prima: Policy-reduced integrity measurement architecture,” in Proceedings of the 2007 ACM workshop on Scalable trusted computing (SACMAT ’06), June 2006.

[9] N. L. Petroni, Jr., T. Fraser, J. Molina, and W. A. Arbaugh. Copilot - a Coprocessor-based Kernel Runtime Integrity Monitor. In Proceedings of the 13th USENIX Security Symposium, 2004.

[10] Jianxin Li, Jinpeng Huai, Chunming Hu, Yanming Zhu. A Secure Collaboration Service for Dynamic Virtual Organizations, Elsevier, Information Sciences, v 180, n 17, p 3086-3107, September 1, 2010.

[11] Google Chrome OS. www.chromium.org/chromium-os

[12] E. M. Stuart and J. D. John, "Application and analysis of the virtual machine approach to information system security and isolation," in Proceedings of the workshop on virtual computer systems. Cambridge, Massachusetts, United States: ACM, 1973.

[13] X. Zhao, K. Borders, and Atul Prakash, Virtual Machine Security Systems,Book Chapter To Appear in Advances in Computer Science and Engineering. , 2009.

[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," 2009.

[15] IKrohn, M., Yip, A., Brodsky, M., Cliffer, N., Kaashoek, M. F., Kohler, E., and Morris, R. 2007. Information flow control for standard OS abstractions. In Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles (Stevenson, Washington, USA, October 14 - 17, 2007). SOSP '07. ACM, New York, NY, 321-334. DOI= http://doi.acm.org/10.1145/1294261.1294293

[16] Yan Wen, Jinjing Zhao, Huaimin Wang, Jiannong Cao: Implicit Detection of Hidden Processes with a Feather-Weight Hardware-Assisted Virtual Machine Monitor. ACISP 2008: 361-375

[17] Ahmed M. Azab, Peng Ning, Emre C. Sezer, and Xiaolan Zhang, "HIMA: A Hypervisor-Based Integrity Measurement Agent," in *Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC '09)*, December 2009, Honolulu, Hawaii, USA.

[18] B. D. Payne and W. Lee, "Secure and Flexible Monitoring of Virtual Machines," presented at 23rd Annual Computer Security Applications Conference (ACSAC), Miami Beach, Florida, USA, 2007.

[19] B. D. Payne, M. Carbone, and M. I. S. W. Lee, "Lares: An Architecture for Secure Active Monitoring Using Virtualization," presented at IEEE Symposium on Security and Privacy (S&P 2008), Oakland, California, USA, 2008.

[20] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: security challenges in virtual machine based computing environments " presented at Proceedings of the 10th conference on Hot Topics in Operating Systems (HOTOS 2005), Berkeley, CA, USA, 2005.

[21] L. Flavio and P. Roberto Di, "KvmSec: a security extension for Linux kernel virtual machines," in Proceedings of the 2009 ACM symposium on Applied Computing. Honolulu, Hawaii: ACM, 2009.

[22] J. Ashlesha, T. K. Samuel, W. D. George, and M. C. Peter, "Detecting past and present intrusions through vulnerability-specific predicates," in Proceedings of the twentieth ACM symposium on Operating systems principles. Brighton, United Kingdom: ACM, 2005.

[23] Amazon Simple Storage Service (Amazon S3), https://s3.amazonaws.com/

[24] David Meisner, Brian T. Gold, and Thomas F. Wenisch. 2009. PowerNap: eliminating server idle power. In Proceeding of the 14th international conference on Architectural support for programming languages and operating systems (ASPLOS '09). ACM, New York, NY, USA, 205-216. DOI=10.1145/1508244.1508269

[25] Kephart, J., Chan, H., Levine, D., Tesauro, G., Rawson, F., Lefurgy, C,: Coordinating multiple autonomic managers to achienve specified power-performance tradeoffs. In: Proc. IEEE Intl.Conf. on Autonomic Computing(ICAC), pp. 145-154, Jun.2007

[26] Kevin Francis and Peter Richardson, Green Maturity Model for Virtualization, http://msdn.microsoft.com/en-us/library/dd393310.aspx

[27] Akshat Verma, Gargi Dasgupta, Tapan Kumar Nayak, Pradipta De, and Ravi Kothari. 2009. Server workload analysis for power minimization using consolidation. In Proceedings of the 2009 conference on USENIX Annual technical conference (USENIX'09). USENIX Association, Berkeley, CA, USA, 28-28.

[28] Jan Stoess, Christian Lang, and Frank Bellosa. 2007. Energy management for hypervisor-based virtual machines. In 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference (ATC'07), Jeff Chase and Srinivasan Seshan (Eds.). USENIX Association, Berkeley, CA, USA, , Article 1 , 14 pages.

[29] í.S. Cunha, I. Viana, J. Palotti, J.M. Almeida, and V. Almeida, "Analyzing security and energy tradeoffs in autonomic capacity management", in Proc. NOMS, 2008, pp.302-309.